Tutorial: Optimization

Cheat Sheet and Practice Problems

ES335 - Machine Learning
IIT Gandhinagar

July 23, 2025

1 Summary from Slides

1.1 Optimization Fundamentals

General Optimization Problem:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$$

Where:

- $f(\boldsymbol{\theta})$: objective function to minimize
- $\boldsymbol{\theta}$: parameter vector
- $\boldsymbol{\theta}^*$: optimal parameter values

Gradient: The gradient $\nabla f(\boldsymbol{\theta})$ points in the direction of steepest ascent:

$$\nabla f(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial f}{\partial \theta_1} \\ \frac{\partial f}{\partial \theta_2} \\ \vdots \\ \frac{\partial f}{\partial \theta_d} \end{bmatrix}$$

1.2 Gradient Descent Algorithm

Basic Algorithm:

1. Initialize $\boldsymbol{\theta}$ to random values
2. Compute gradient $\nabla f(\boldsymbol{\theta})$
3. Update: $\boldsymbol{\theta}_{i+1} \leftarrow \boldsymbol{\theta}_i - \alpha \nabla f(\boldsymbol{\theta}_i)$
4. Repeat until convergence

Key Properties:

- First-order optimization method (uses gradients only)
- Iterative algorithm
- Local search (greedy approach)
- Learning rate α controls step size

1.3 Taylor Series Foundation

First-order approximation:

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0)$$

Second-order approximation:

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T\nabla^2 f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

Where $\nabla^2 f(\mathbf{x}_0)$ is the Hessian matrix of second derivatives.

1.4 Convexity

Convex Function: A function f is convex if for any $\mathbf{x}, \mathbf{y}$ and $\lambda \in [0, 1]$:

$$f(\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y})$$

Examples of Convex Functions:

- $f(x) = x^2$ (quadratic)
- $f(x) = |x|$ (absolute value)
- $f(x) = e^x$ (exponential)
- $f(x) = -\log(x)$ for $x > 0$

Properties:

- Convex functions have unique global minimum
- Any local minimum is a global minimum
- Gradient descent converges to global optimum

1.5 Coordinate Descent

Algorithm: Optimize one coordinate at a time while keeping others fixed

Steps:

1. Pick coordinate j (random or cyclic)
2. Solve: $\theta_j^* = \arg\min_{\theta_j} f(\theta_1, \ldots, \theta_j, \ldots, \theta_d)$
3. Update $\theta_j \leftarrow \theta_j^*$
4. Repeat until convergence

Advantages:

- No step-size selection needed
- Each update solves a 1D optimization problem
- Effective for certain problems (e.g., Lasso)

1.6 Subgradients

For non-differentiable convex functions, use **subgradients**:

Definition: g is a subgradient of f at x_0 if:

$$f(x) \geq f(x_0) + g^T(x - x_0) \quad \forall x$$

Example: For $f(x) = |x|$:

$$\frac{\partial |x|}{\partial x} = \begin{cases} 1 & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

1.7 Advanced Methods

Stochastic Gradient Descent (SGD):

$$\boldsymbol{\theta}_{i+1} \leftarrow \boldsymbol{\theta}_i - \alpha \nabla f_i(\boldsymbol{\theta}_i)$$

where f_i is the loss for a single example or mini-batch.

Advantages: Faster updates, better for large datasets **Disadvantages**: Noisy updates, may oscillate around optimum

2 Practice Problems

Problem : Basic Gradient Calculation

For $f(x, y) = x^2 + 2y^2 + xy$:
a) Calculate the gradient $\nabla f(x, y)$ b) Evaluate the gradient at point $(1, 2)$ c) In which direction should you move to decrease f most rapidly?

Problem : Gradient Descent Steps

Starting with $\boldsymbol{\theta}_0 = [1, 1]^T$ and learning rate $\alpha = 0.1$, perform 2 iterations of gradient descent on $f(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2$.
a) Calculate $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ b) Compute $f(\boldsymbol{\theta}_0)$, $f(\boldsymbol{\theta}_1)$, and $f(\boldsymbol{\theta}_2)$ c) Verify that the function value decreases

Problem : Learning Rate Analysis

For $f(x) = x^2$ starting at $x_0 = 4$, analyze gradient descent with different learning rates:
a) $\alpha = 0.1$: Calculate x_1, x_2, x_3 b) $\alpha = 1.0$: Calculate x_1, x_2, x_3 c) $\alpha = 2.1$: Calculate x_1, x_2, x_3 d) What happens in each case? When does the algorithm diverge?

Problem : Convexity Verification

Determine if the following functions are convex:
a) $f(x) = x^4$ b) $f(x) = \sin(x)$ c) $f(x) = \max(0, x)$ (ReLU function) d) $f(x, y) = x^2 + y^2 - xy$
For each, either prove convexity or provide a counterexample.

Problem : Taylor Series Approximation

For $f(x) = e^x$ around $x_0 = 0$:
a) Write the first-order Taylor approximation b) Write the second-order Taylor approximation c) Evaluate both approximations at $x = 0.1$ and compare with the true value d) How does the approximation quality change as you move away from x_0?

Problem : Coordinate Descent Implementation

Apply coordinate descent to minimize $f(\theta_1, \theta_2) = \theta_1^2 + 2\theta_2^2 + \theta_1\theta_2$ starting from $(2, 1)$:
a) Fix $\theta_2 = 1$ and minimize over θ_1 b) Fix θ_1 to the new value and minimize over θ_2 c) Compare this to one step of gradient descent with $\alpha = 0.5$ d) Which method makes more progress toward the minimum?

Problem : Subgradient Calculation

For $f(x) = |x - 2| + |x + 1|$:
a) Find the subgradient at $x = 2$ b) Find the subgradient at $x = -1$ c) Find the subgradient at $x = 0$ d) Where is the minimum of this function?

Problem : Linear Regression Optimization

For linear regression with $f(\boldsymbol{\theta}) = \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2$:
a) Derive the gradient $\nabla f(\boldsymbol{\theta})$ b) Write the gradient descent update rule c) Show that the optimal solution is $\boldsymbol{\theta}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ d) When would you prefer gradient descent over the closed-form solution?

Problem : Convergence Criteria

Design stopping criteria for gradient descent:
a) Gradient magnitude: $||\nabla f(\boldsymbol{\theta})|| < \epsilon$ b) Parameter change: $||\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k|| < \delta$ c) Function value change: $|f(\boldsymbol{\theta}_{k+1}) - f(\boldsymbol{\theta}_k)| < \gamma$
For each criterion: - When might it fail to detect convergence? - What are appropriate threshold values? - How do they perform on flat vs. steep functions?

Problem : Stochastic Gradient Descent

Compare batch gradient descent and SGD for minimizing the average of functions:

$$f(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta})$$

a) Write the update rules for both methods b) What are the computational costs per iteration? c) How does the noise in SGD affect convergence? d) When is SGD preferred over batch gradient descent?

Problem : Local vs Global Minima

Consider $f(x) = x^4 - 4x^3 + 4x^2$:
a) Find all critical points by setting $f'(x) = 0$ b) Classify each critical point (local min, local max, saddle) c) Starting from different initial points, where would gradient descent converge? d) How could you modify the algorithm to escape local minima?

Problem : Momentum Methods

Standard momentum updates parameters as:

$$\mathbf{v}_{k+1} = \beta \mathbf{v}_k + \alpha \nabla f(\boldsymbol{\theta}_k)$$

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mathbf{v}_{k+1}$$

a) How does momentum help with oscillations? b) What happens when $\beta = 0$ (no momentum)? c) What happens when $\beta \to 1$ (high momentum)? d) Apply momentum gradient descent to $f(x, y) = x^2 + 10y^2$ and observe the path.

Problem : Newton's Method

Newton's method uses second-order information:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - (\nabla^2 f(\boldsymbol{\theta}_k))^{-1} \nabla f(\boldsymbol{\theta}_k)$$

a) Apply Newton's method to $f(x) = x^2 - 4x + 3$ starting from $x_0 = 0$ b) Compare convergence rate with gradient descent c) What are the computational advantages and disadvantages? d) Why is Newton's method rarely used in machine learning?

Problem : Constrained Optimization Setup

Convert the constrained problem to unconstrained using Lagrangian:
Minimize: $f(x, y) = x^2 + y^2$ Subject to: $g(x, y) = x + y - 1 = 0$
a) Write the Lagrangian function b) Find the KKT conditions c) Solve for the optimal (x^*, y^*, λ^*) d) Verify that the constraint is satisfied

Problem : Adaptive Learning Rates

AdaGrad adapts learning rates based on historical gradients:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \frac{\alpha}{\sqrt{G_k + \epsilon}} \odot \nabla f(\boldsymbol{\theta}_k)$$

Where G_k accumulates squared gradients.
a) Explain why this helps with sparse gradients b) What problem does the ϵ term solve? c) Why might AdaGrad's learning rate become too small over time? d) How do newer methods like Adam address this issue?

Problem : Real-world Optimization Strategy

Design an optimization strategy for training a neural network on image classification:
Dataset: 1M images, 1000 classes Network: 50 million parameters
a) Which optimization algorithm would you choose and why? b) How would you set the initial learning rate? c) Design a learning rate schedule for training d) What techniques would you use to avoid overfitting? e) How would you monitor and debug training progress?